Universidad Central “Marta Abreu” de Las Villas

Proyecto Final de Ingeniería Automática



Tema: **Implementación de un Micro PLC utilizando Arduino IDE y el Microcontrolador ESP32 para Aplicaciones de IoT.**

Integrantes:

César Gabriel García Pérez #7.

Hainer Jonás González Díaz #8.

Alberto Eduardo Cubela Trujillo #4.

Introducción:

En el siglo XXI, el desarrollo de proyectos de ingeniería en automática es crucial, ya que todos los centros y lugares donde se emplea la automática utilizan PLC (Controladores Lógico Programables). Estos dispositivos, producidos por firmas reconocidas a nivel internacional como Schneider Electric y Siemens, son de una calidad excepcional, pero tienen precios elevados. Sin embargo, las aplicaciones que queremos desarrollar se enmarcan en ambientes donde no se necesitan dispositivos tan robustos y costosos.

Este proyecto tiene como objetivo la implementación de un micro PLC utilizando Arduino IDE y el microcontrolador ESP32. Este micro PLC se instalará en habitaciones hoteleras o incluso en casas de cultivo, donde se requiere la automatización de máquinas de riego de pivote central, entre otras aplicaciones. La arquitectura propuesta es más sencilla y económica, resolviendo de igual forma los problemas diarios.

**Objetivos del Proyecto**

**Objetivo General:**  
Implementar un micro PLC utilizando Arduino IDE y el microcontrolador ESP32 para automatizar procesos en entornos como habitaciones de hotel y casas de cultivo.

**Objetivos Específicos:**

1. Investigar las características y ventajas del ESP32 en comparación con PLC tradicionales.
2. Desarrollar el software necesario para la programación del micro PLC utilizando Arduino IDE.
3. Instalar el micro PLC en un entorno controlado, como una habitación hotelera o un sistema de riego en una casa de cultivo.
4. Realizar pruebas funcionales del sistema para asegurar su operatividad y efectividad.
5. Evaluar los resultados obtenidos durante las pruebas y ajustar el sistema según sea necesario.

Desarrollo:

**Marco teórico relevante:**

1. Introducción a la Automatización en el Sector Hotelero

La automatización en el sector hotelero ha evolucionado significativamente, impulsada por la necesidad de mejorar la eficiencia operativa y la experiencia del huésped. Los sistemas automatizados permiten un control más preciso sobre las condiciones ambientales, optimizando el uso de recursos como energía y agua. La implementación de tecnologías IoT (Internet de las Cosas) ha facilitado esta transición, permitiendo la integración de dispositivos conectados que pueden ser monitoreados y controlados de forma remota.

2. Tecnologías IoT en Hoteles Inteligentes

Los hoteles inteligentes utilizan una variedad de tecnologías para ofrecer servicios automatizados. Estas incluyen:

* Sensores: Dispositivos que recopilan datos sobre temperatura, humedad, movimiento y apertura de puertas/ventanas. Estos sensores son esenciales para crear un ambiente confortable y seguro para los huéspedes.
* Controladores Lógicos Programables (PLC): Aunque tradicionalmente se utilizan PLC industriales, el uso de microcontroladores como el ESP32 permite una solución más económica y flexible para la automatización en entornos hoteleros.
* Relés: Permiten controlar dispositivos eléctricos como luces y sistemas de climatización, facilitando la automatización basada en las condiciones detectadas por los sensores.

3. Beneficios de la Automatización

La implementación de un sistema automatizado en habitaciones hoteleras ofrece múltiples beneficios:

* Confort del Huésped: La capacidad de ajustar automáticamente la temperatura y la iluminación mejora la experiencia del huésped.
* Eficiencia Energética: Al optimizar el uso de recursos, se reduce el consumo energético, lo que resulta en ahorros significativos para el hotel.
* Mantenimiento Proactivo: Sensores que detectan problemas (como ventanas abiertas) pueden alertar al personal sobre situaciones que podrían afectar la eficiencia energética o la seguridad.

**Materiales y Métodos**

**Placa de Desarrollo:**

* LilyGo T3 S3: Microcontrolador con capacidades Wi-Fi y Bluetooth, ideal para aplicaciones IoT.

**Sensores:**

* Sensor de Temperatura y Humedad (DHT11 o DHT22): Para monitorear las condiciones ambientales.
* Sensor de Movimiento (PIR): Para detectar la presencia de huéspedes en la habitación.
* Sensor de Apertura de Puertas y Ventanas: Para asegurarse de que el aire acondicionado se apague si se detecta que están abiertas.

**Relés:**

* Módulos de Relé: Para controlar dispositivos eléctricos como el aire acondicionado y las luces.

**Cables y Conectores:**

* Cables jumper para realizar las conexiones entre los componentes.

**Fuente de Alimentación:**

* Fuente adecuada para alimentar la placa y otros dispositivos conectados.

**Software:**

* Arduino IDE: Entorno de desarrollo para programar la lógica del sistema en C++.

**Opcional:**

* Cámara (ESP32-CAM): Para implementar funcionalidades adicionales como monitoreo visual.
* Módulos adicionales (si es necesario): Como sensores de calidad del aire o sistemas de control remoto.

**Métodos**

**Selección de Hardware y Software:**

Basándose en criterios como costo, disponibilidad y compatibilidad con la placa LilyGo T3 S3 se seleccionaron los componentes adecuados (sensores, relés, microcontroladores)

**Programación:**

Se utiliza el lenguaje C++ en el entorno Arduino IDE para desarrollar el código que controlará los dispositivos según las entradas de los sensores.

El código incluiye funciones para:

Leer datos de temperatura y humedad.

Detectar movimiento y controlar el encendido/apagado de luces.

Monitorear la apertura de puertas y ventanas para gestionar el aire acondicionado.

**Montaje del Sistema:**

Realizar las conexiones físicas entre la placa LilyGo T3 S3, los sensores, y los módulos de relé siguiendo un esquema eléctrico claro.

**Pruebas y Validación:**

Realizar pruebas exhaustivas para verificar que cada sensor funcione correctamente y que el sistema responda adecuadamente a las condiciones detectadas.

Evaluar el rendimiento del sistema bajo diferentes escenarios (por ejemplo, cambios en temperatura, presencia de personas, etc.) para asegurar su efectividad.

**Descripción del Empleo de la Placa para Automatizar una Habitación de Hotel**

En el contexto del control de un entorno IoT para una habitación hotelera, se han discutido varias preguntas clave relacionadas con la implementación de sensores y el uso de la placa de desarrollo LilyGo T3 S3. A continuación, se presenta un resumen:

**Sensores**

Es sugerente la utilización de sensores de temperatura, humedad, movimiento, y apertura de puertas y ventanas. Estos dispositivos son fundamentales para monitorear y controlar el ambiente de la habitación, asegurando el confort del huésped y optimizando el uso de energía.

**Control de Relés**

Mediante la implementación de relés se pueden manejar dispositivos como el aire acondicionado y las luces. Para hacer un proyecto competente, el sistema debe ser capaz de:

1. Mantener una temperatura óptima cuando no hay presencia en la habitación.
2. Apagar el aire acondicionado si se detecta que las ventanas o puertas están abiertas durante un tiempo prolongado.
3. Encender las luces automáticamente al detectar presencia en la habitación.

**Código de Implementación**

A continuación, se presenta un ejemplo de código en C++ que utiliza la placa LilyGo T3 S3 para integrar los sensores mencionados y controlar los relés según las condiciones detectadas. Este código permite leer datos de los sensores y tomar decisiones automatizadas sobre el funcionamiento del aire acondicionado y las luces.

***Ejemplo de código:***

#include <WiFi.h>

#include <PubSubClient.h>

#include <DHT.h>

// Configuración de WiFi

const char\* ssid = "YOUR\_SSID";

const char\* password = "YOUR\_PASSWORD";

// Configuración de MQTT

const char\* mqttServer = "MQTT\_BROKER\_IP";

const int mqttPort = 1883;

const char\* mqttUser = "MQTT\_USER";

const char\* mqttPassword = "MQTT\_PASSWORD";

WiFiClient espClient;

PubSubClient client(espClient);

// Configuración del sensor DHT

#define DHTPIN 4 // Pin donde está conectado el DHT

#define DHTTYPE DHT11 // Tipo de sensor DHT (DHT11 o DHT22)

DHT dht(DHTPIN, DHTTYPE);

// Pines para relés

#define RELAY\_AC 5 // Relé para aire acondicionado

#define RELAY\_LIGHT 6 // Relé para luces

#define SENSOR\_DOOR 7 // Sensor de puerta (abierto/cerrado)

#define SENSOR\_WINDOW 8 // Sensor de ventana (abierto/cerrado)

// Variables para almacenar datos

float temperature;

float humidity;

bool presenceDetected = false; // Variable para detectar presencia

void setup() {

Serial.begin(115200);

dht.begin();

pinMode(RELAY\_AC, OUTPUT);

pinMode(RELAY\_LIGHT, OUTPUT);

pinMode(SENSOR\_DOOR, INPUT);

pinMode(SENSOR\_WINDOW, INPUT);

setup\_wifi();

client.setServer(mqttServer, mqttPort);

}

void setup\_wifi() {

delay(10);

Serial.println();

Serial.print("Conectando a ");

Serial.println(ssid);

WiFi.begin(ssid, password);

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

Serial.println("");

Serial.println("Conectado a WiFi");

}

void reconnect() {

while (!client.connected()) {

Serial.print("Intentando conexión MQTT...");

if (client.connect("LilyGo\_T3\_S3", mqttUser, mqttPassword)) {

Serial.println("conectado");

client.subscribe("hotel/room/control");

} else {

Serial.print("falló, rc=");

Serial.print(client.state());

delay(2000);

}

}

}

void readSensors() {

humidity = dht.readHumidity();

temperature = dht.readTemperature();

if (isnan(humidity) || isnan(temperature)) {

Serial.println("Error al leer el sensor DHT");

return;

}

// Controlar aire acondicionado

if (presenceDetected) {

if (digitalRead(SENSOR\_DOOR) == HIGH || digitalRead(SENSOR\_WINDOW) == HIGH) {

digitalWrite(RELAY\_AC, LOW); // Apagar aire acondicionado

Serial.println("Aire acondicionado apagado por ventana/puerta abierta.");

} else {

if (temperature > 24.0) { // Temperatura de confort

digitalWrite(RELAY\_AC, HIGH); // Encender aire acondicionado

Serial.println("Aire acondicionado encendido.");

} else {

digitalWrite(RELAY\_AC, LOW); // Apagar aire acondicionado

Serial.println("Aire acondicionado apagado.");

}

}

} else {

digitalWrite(RELAY\_AC, LOW); // Apagar aire si no hay presencia

Serial.println("Aire acondicionado apagado por falta de presencia.");

}

// Controlar luces

if (presenceDetected) {

digitalWrite(RELAY\_LIGHT, HIGH); // Encender luces

Serial.println("Luces encendidas.");

} else {

digitalWrite(RELAY\_LIGHT, LOW); // Apagar luces

Serial.println("Luces apagadas.");

}

}

void loop() {

if (!client.connected()) {

reconnect();

}

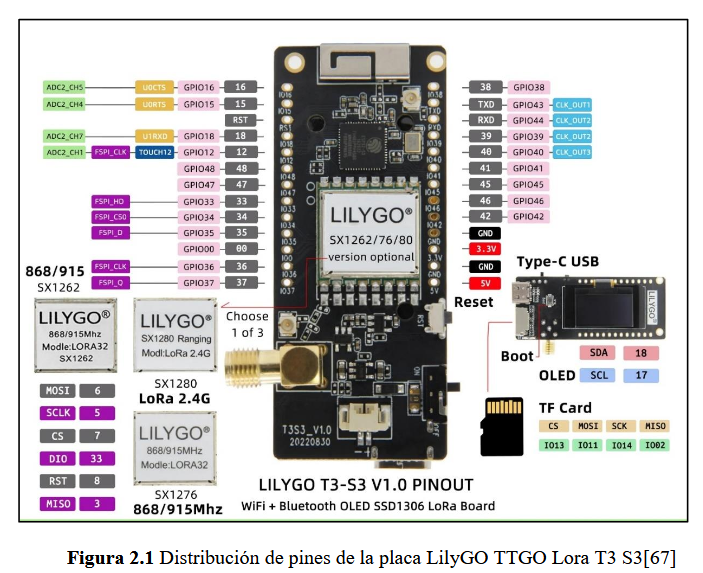
client.loop();

readSensors(); // Leer y controlar cada vez que se ejecuta el loop

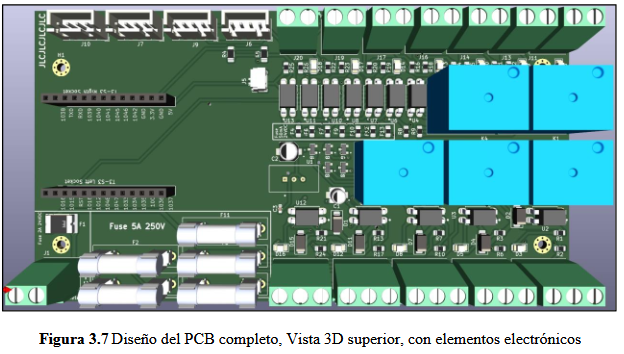
delay(5000); // Esperar 5 segundos entre lecturas

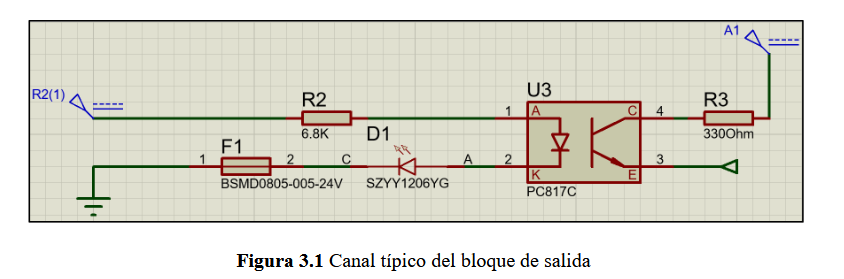
}

**Descripción de la PCB y componentes electrónicos, sus principales datos técnicos:**

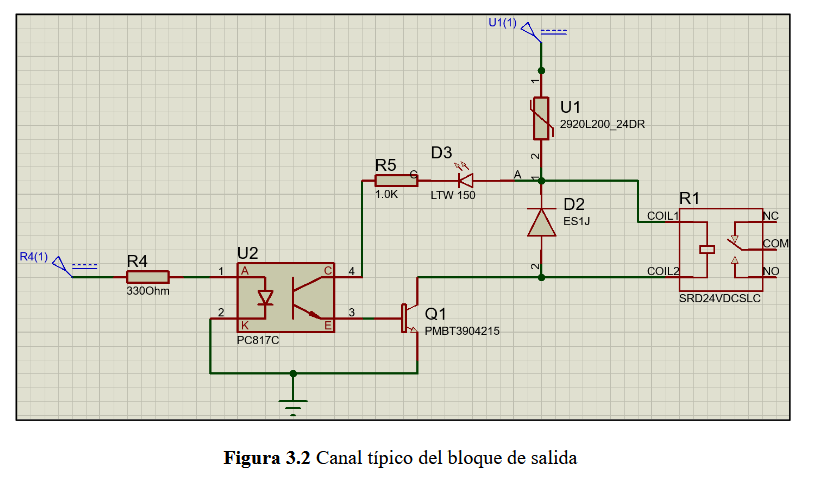
****

La placa LilyGo T3 S3 contiene el microcontrolador ESP32, que actúa como el cerebro del micro PLC. En la imagen se puede apreciar el pinout del dispositivo para su posterior empleo.

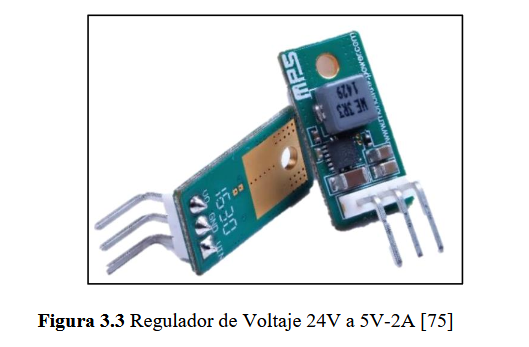


Partiendo de los sistemas de entrada/salida especificados, se enumeran 7 entradas y 5 salidas. Estos bloques de entrada-salida comprenden dos diseños típicos que se repiten para ambos procesos. La Figura 3.1 representa el canal de entrada, simulado en Proteus. Dado que se tendrán 7 canales en el circuito electrónico de la PCB, cada canal está equipado con optoacopladores para desacoplar ruidos o aislar las variaciones de voltaje que puedan surgir debido a interferencias. La señal de entrada que llega es de 24VDC. Dado que es una señal de 24 voltios, se introduce una resistencia de 6.8K ohm para regular la corriente que llega al optoacoplador. Además, se incorpora un fusible de protección de 50mA y un LED para notificar visualmente que la entrada está activada. Este circuito se conecta hasta un pin de entrada previamente descrito hasta la T3 S3.

La Figura 3.2 representa el canal de salida simulado en Proteus, y se replicaron 5 canales idénticos en el circuito electrónico de la PCB. Este canal está equipado con un relé que se utiliza para accionar contactores magnéticos responsables del control de elementos dentro de la habitación de hotel. Para lograr esta acción, se emplea un optoacoplador que se activa cuando el pin de salida envía una señal de 3.3 voltios. Del otro lado, permite el paso de los 24VDC hasta el transistor, el cual se excita y permite el flujo de corriente a través del bobinado del relé. Con el objetivo de proteger el circuito, se incorpora el diodo D2, que cumple dos funciones cruciales: protección contra retroalimentación del relé debido a la energía almacenada en la bobina del mismo y supresión de picos de voltaje. Además, se utiliza un diodo LED para indicar el estado de encendido del magnético en relación con el circuito de salida. Se incluyen resistencias para limitar altas corrientes por el circuito, y se agrega un fusible que regula la entrada a toda la placa hasta 2A de corriente. Este circuito se conecta hasta un pin de salida previamente descrito en la Tabla 2.2 hasta la T3 S3.

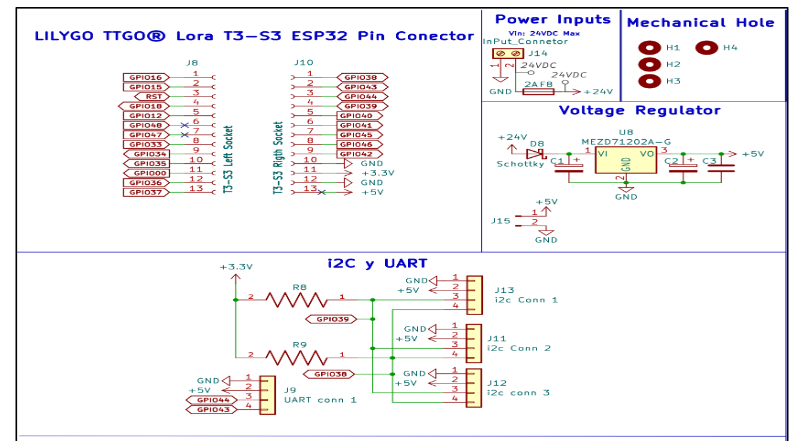


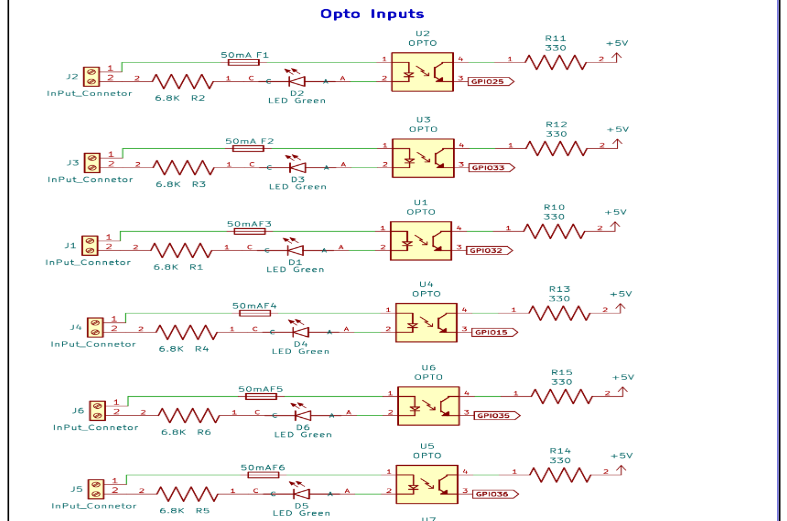
La fuente utilizada en este diseño es una de 24 VDC proveniente del exterior, que alimenta todo el bloque de salida, incluyendo los relés de potencia y el regulador de voltaje MEZD71202A-G. Este regulador se utiliza para estabilizar el voltaje de entrada de 24VDC, aspecto crucial para el funcionamiento de los optoacopladores U1 hasta U7 y la alimentación de la placa IoT T3 S3. El regulador acepta un rango de voltaje desde 4.5VDC hasta 24VDC y suministra una salida fija de 5VDC, con una capacidad máxima de corriente de hasta 2A. Se monta en la PCB a través de orificios (TH) y se conecta mediante 3 pines, como se ilustra en la Figura 3.3. En la placa, el regulador está acompañado de tres capacitores que contribuyen al completo filtrado de la entrada y salida de corriente del regulador, así se suprime la gran mayoría de los ruidos eléctricos. Con el fin de garantizar el buen funcionamiento de la placa IoT T3 S3, alimentada por este regulador, se añadió un diodo en la entrada para asegurar que la polaridad siempre sea correcta.

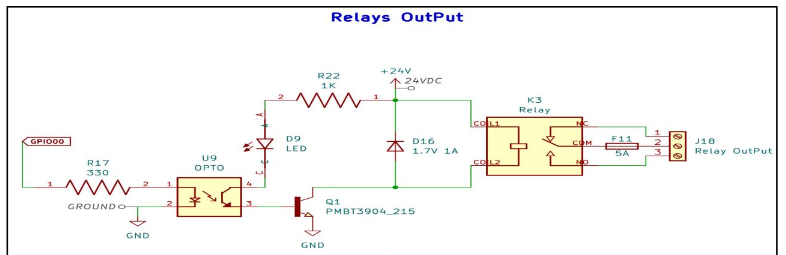


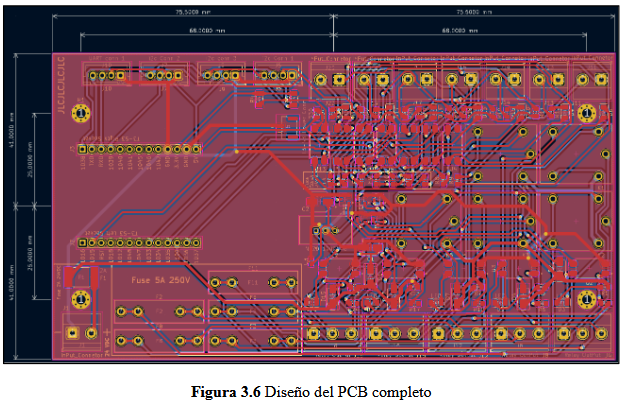
**Diseño en KiCAD:**

El esquemático presentado fue diseñado en la zona de trabajo de KiCAD y sirve como base para la creación del PCB. En él, se pueden identificar todos los bloques de entrada y salida previamente especificados, conectados a sus respectivos pines como se describe en la Tabla 2.1 y Tabla 2.2. Además de estos bloques, se incluyen bloques adicionales. El primer bloque, etiquetado como "LilyGo TTGO Lora T3 S3 ESP32 Pin Conector", representa las salidas y entradas de los pines de conexión de la T3 S3 en la placa. El siguiente bloque, llamado Power input, se refiere a la entrada de 24VDC con su correspondiente fusible reiniciable de 24VDC y 2A, diseñado para proteger el circuito. También se presenta el bloque Voltaje Regulator, explicado anteriormente, incluyendo el conector J15 para alimentar la placa IoT desde ese puerto. También aparece un bloque con los protocolos de comunicación I2C y UART, para instalar diferentes sensores al PCB.









**Mejoras Futuras**

Se sugiere que el sistema podría expandirse para incluir más funcionalidades, como la integración con sistemas de gestión hotelera o la capacidad de enviar notificaciones al personal del hotel sobre el estado de las habitaciones.

Resultados:

Durante la implementacion del codigo en el hardware se detectaron varios problemas:

1. El digrama de entrada-salida de la ESP32 LilyGo T3-S3 no funcionaba sobre los esquemas analogicos de la PCB como en la documetacion obtenida.
2. Las salidas del microcontrolador solo hacen conmutar uno de los reles de salida.
3. La unica manera de conmutar una salida fue manualmente, poniendo una de las salidas del microcontrolador directamente a 5V.

Conclusiones:

Al terminar con la realización de este trabajo final de Ingeniería Automática, se ha podido concluir la importancia del estudio de otras posibilidades de creación y diseño de PCB con la funcionalidad de micro PLC para ser empleados en una amplia gama de servicios donde se necesite la supervisión y control de tareas y procesos, garantizando no solo una calidad excelente sino también un precio mucho más módico en comparación a los PLC que ofrecen firmas reconocidas en el mercado internacional. Se han afianzado más las habilidades en el uso de la herramienta KiCAD para el diseño de placas electrónicas con prestaciones específicas para cada proyecto que se proponga realizar, y se ha aprendido más sobre el entorno de desarrollo integrado de Arduino y sobre la programación específicamente de los microcontroladores ESP32.

### Recomendaciones

1. **Expansión de Funcionalidades**: Se recomienda expandir el sistema para incluir más funcionalidades, como la integración con sistemas de gestión hotelera. Esto permitiría una administración centralizada y eficiente de las habitaciones del hotel.
2. **Notificaciones y Alertas**: Implementar la capacidad de enviar notificaciones y alertas al personal del hotel sobre el estado de las habitaciones. Esto podría incluir alertas sobre ventanas o puertas abiertas, fallos en el sistema de aire acondicionado, o cualquier otra condición que requiera atención inmediata.
3. **Pruebas y Validación**: Realizar pruebas exhaustivas y validación del sistema en diferentes entornos para asegurar su robustez y fiabilidad. Esto incluye pruebas de estrés, pruebas de compatibilidad y pruebas de seguridad.
4. **Documentación y Soporte**: Desarrollar una documentación detallada y proporcionar soporte técnico para facilitar la implementación y el mantenimiento del sistema.
5. **Actualizaciones y Mantenimiento**: Asegurar que el sistema se mantenga actualizado y en optimo funcionamiento.